# MATH4100-IntelyCare Project Report

Yihao Li
Marcel Almonte
Wesley Le
James Voss

Instructor: Semere Habtemicael
Wentworth Institute of Technology

## 1 Abstract

Intelycare is a platform where nurses can work available shifts at participating long-term care facilities as they choose. They give nursing professionals full flexibility to work as much or as little as they choose. Thus, the goal is to use unsupervised learning techniques to identify cohorts of nurses with similar in-app behaviors. Our goal is to using Semi-Supervised Machine Learning to label the data set, to classify the group of nurse by clustered data, and to predict several nurse behaviors. The techniques and models we used in this study is PCA, K-mean clustering, DBSCAN clustering, logistic regression, LDA, and Trees.

## 2 Introduction

IntelyCare uses data and technology to help long-term care facilities find the extra help they need on a per-shift basis from our pool of credentialed nursing professionals. We give nursing professionals full flexibility to work as much or as little as they choose, and we observe substantial variance in usage across our pool of nurses. Our goal is to identify cohorts of nurses with similar in-app behaviors. Who are nurses that accept whatever shift we show them? Who are the nurses who only work when they find the exact shift they like? Which nurses can work more shifts in a week and which nurses cannot? Who are the weekend warriors? Who are the holiday heroes?

These behavioral cohorts will allow us to personalize the nurse experience in important ways. We can send more timely and relevant communications, improve our in-app shift recommendations, and offer more effective promotions if we understand what behavior patterns characterize a given nursing professional.

To achieve this goal, we used semi-supervised machine learning techniques. In general, problems where you have a large amount of input data (X) and only some of the data is labeled (Y) are called semi-supervised learning problems. These problems sit in between both supervised and unsupervised learning.

### 2.1 Data Visualization

For data visualization, we used multidimensional scaling.

#### 2.1.1 Multidimensional Scaling

Multidimensional Scaling (MDS) is a way to present data to show similarity between individual cases within a data set. It creates new variables that still represent the original variables, as well as make it so that the data can be represented in a 2-D display with N dimensions, highlighting the difference, or distance, between the points.

### 2.2 Unsupervised learning

Unsupervised learning, also known as unsupervised machine learning, uses machine learning algorithms to analyze and cluster unlabeled datasets. These algorithms discover hidden patterns or data groupings without the need for human intervention. Its ability to discover similarities and differences in information make it

the ideal solution for exploratory data analysis, cross-selling strategies, customer segmentation, and image recognition.

### 2.2.1 Principal Component Analysis

In the research and application of many fields, it is necessary to observe the data containing multiple variables, collect a large amount of data and analyze it to find the law. Multivariate large data sets will undoubtedly provide rich information for research and application, but it also increases the workload of data collection to a certain extent. More importantly, in many situations, there may be correlations between many variables, which increases the complexity of problem analysis. If you analyze each indicator separately, the analysis is often isolated and cannot fully utilize the information in the data. Therefore, blindly reducing the indicators will lose a lot of useful information, resulting in wrong conclusions.

Therefore, it is necessary to find a reasonable method to minimize the loss of information contained in the original indicators while reducing the indicators that need to be analyzed, to achieve the purpose of comprehensive analysis of the collected data. Since there is a certain correlation between the variables, it can be considered to change the closely related variables into as few new variables as possible, so that these new variables are pairwise uncorrelated, then fewer comprehensive indicators can be used to represent them, respectively. Various types of information that exist in various variables. Principal component analysis and factor analysis belong to this type of dimensionality reduction algorithm. PCA is an exploratory tool to simplify a large and complex data set into a smaller, more easily understandable data set. It summarizes the complex data set by creating variables that are linear combinations of the original data set.

It does not necessarily describe or explain the relationship among variables. After constructing principal component analysis, each of new variables, called principal components, is uncorrelated with all others. It is effectively used as a method of pattern analysis bringing strong patterns to the forefront. In summary, PCA looks to find a low-dimensional representation of the observation that explain a good fraction of the variance. The other technique is clustering. There are two clustering techniques discussed in this book.

### 2.2.2 Clustering Methods

Depending on the clustering method that is used, there are certain criterias that determine the output.K-Means and spectral clustering is a method of clustering where the user is able to determine the number of desired clusters. There are many others, but this is getting long. Reference the clustering resource for the full list.

- **K-mean Clustering**
  K-means is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed apriori. The main idea is to define k centers, one for each cluster. These centers should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest center. When no point is pending, the first step is completed and an early group age is done. At this point we need to re-calculate k new centroids as barycenter of theclusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new center. A loop has been generated. As a result of this loop we may notice that the k centers change their location step by step until no more changes are done or in other words centers do not move any more. Finally, this algorithm aims at minimizing an objective function know as squared error function given by:

$$J(V) = \sum_{i=1}^{c} \sum_{j=1}^{c_i} \left( \|x_i - v_j\| \right)^2$$

  The algorithmic steps for k-mean clustering is shown with following: Let $X = (x1, x2, x3, \ldots\ldots\ldots, x_n)$ be the set of data points and V = ($v_1$,$v_2$,.....,$v_c$) be the set of centers.1) Randomly select 'c' cluster centers. 2) Calculate the distance between each data point and cluster centers. 3) Assign the data point to the cluster center whose distance from the cluster center is minimum of all the cluster centers. 4) Recalculate the new cluster center using: where, 'ci' represents the number of data points in $i_th$

cluster.

$$\boldsymbol{v}_i = (1/c_i) \sum_{j=1}^{\mathcal{C}_i} \boldsymbol{x}_i$$

5) Recalculate the distance between each data point and new obtained cluster centers. 6) If no data point was reassigned then stop, otherwise repeat from step 3).

- **Density based clustering algorithm**
  The DBSCAN algorithm views clusters as areas of high density separated by areas of low density. Due to this rather generic view, clusters found by DBSCAN can be any shape, as opposed to k-means which assumes that clusters are convex shaped. The central component to the DBSCAN is the concept of core samples, which are samples that are in areas of high density. A cluster is therefore a set of core samples, each close to each other (measured by some distance measure) and a set of non-core samples that are close to a core sample (but are not themselves core samples). There are two parameters to the algorithm, min samples and eps, which define formally what we mean when we say dense. Higher min samples or lower eps indicate higher density necessary to form a cluster.

  In genereal, Density based clustering algorithm has played a vital role in finding non linear shapes structure based on the density. Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is most widely used density based algorithm. It uses the concept of density reachability and density connectivity. Density Reachability - A point "p" is said to be density reachable from a point "q" if point "p" is within $\mathcal{E}$ distance from point "q" and "q" has sufficient number of points in its neighbors which are within distance $\mathcal{E}$. Density Connectivity - A point "p" and "q" are said to be density connected if there exist a point "r" which has sufficient number of points in its neighbors and both the points "p" and "q" are within the $\mathcal{E}$ distance. This is chaining process. So, if "q" is neighbor of "r", "r" is neighbor of "s", "s" is neighbor of "t" which in turn is neighbor of "p" implies that "q" is neighbor of "p". The algorithmic steps for DBSCAN clustering algorithm can be shown as follows: Let $X = x1, x2, x3, ..., x_n$ be the set of data points. DBSCAN requires two parameters: $\mathcal{E}$ (eps) and the minimum number of points required to form a cluster (minPts). 1) Start with an arbitrary starting point that has not been visited. 2) Extract the neighborhood of this point using $\mathcal{E}$ (All points which are within the $\mathcal{E}$ distance are neighborhood). 3) If there are sufficient neighborhood around this point then clustering process starts and point is marked as visited else this point is labeled as noise (Later this point can become the part of the cluster). 4) If a point is found to be a part of the cluster then its $\mathcal{E}$ neighborhood is also the part of the cluster and the above procedure from step 2 is repeated for all $\mathcal{E}$ neighborhood points. This is repeated until all points in the cluster is determined. 5) A new unvisited point is retrieved and processed, leading to the discovery of a further cluster or noise. 6) This process continues until all points are marked as visited.

## 2.3 Supervised Learning

Supervised learning is a type of machine learning task. It derives the prediction function from the labeled training data. Labeled training data means that each training instance includes input and expected output. When dealing with supervised data the goal is to infer from the training set by means of regression and classification. These regressions can be linear or polynomial. In terms of classification one can classify data by means of logistic regression, k nearest neighbors, and decision trees. When it comes to regression the purpose is to predict numerical values, yet in terms of classification the purpose is to predict categorical values.

### 2.3.1 Logistic Regression

While linear regression is leveraged when dependent variables are continuous, logistical regression is selected when the dependent variable is categorical, meaning they have binary outputs, such as "true" and "false" or "yes" and "no." While both regression models seek to understand relationships between data inputs, logistic regression is mainly used to solve binary classification problems, such as spam identification.

#### 2.3.2 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is most commonly used as dimensionality reduction technique in the pre-processing step for pattern-classification and machine learning applications. The goal is to project a dataset onto a lower-dimensional space with good class-separability in order avoid overfitting ("curse of dimensionality") and also reduce computational costs. Linear discriminant analysis (LDA), normal discriminant analysis (NDA), or discriminant function analysis is a generalization of Fisher's linear discriminant, a method used in statistics and other fields, to find a linear combination of features that characterizes or separates two or more classes of objects or events. The resulting combination may be used as a linear classifier, or, more commonly, for dimensionality reduction before later classification. LDA is closely related to analysis of variance (ANOVA) and regression analysis, which also attempt to express one dependent variable as a linear combination of other features or measurements. However, ANOVA uses categorical independent variables and a continuous dependent variable, whereas discriminant analysis has continuous independent variables and a categorical dependent variable (i.e. the class label). Logistic regression and probit regression are more similar to LDA than ANOVA is, as they also explain a categorical variable by the values of continuous independent variables. These other methods are preferable in applications where it is not reasonable to assume that the independent variables are normally distributed, which is a fundamental assumption of the LDA method.LDA is also closely related to principal component analysis (PCA) and factor analysis in that they both look for linear combinations of variables which best explain the data. LDA explicitly attempts to model the difference between the classes of data. PCA, in contrast, does not take into account any difference in class, and factor analysis builds the feature combinations based on differences rather than similarities. Discriminant analysis is also different from factor analysis in that it is not an interdependence technique: a distinction between independent variables and dependent variables (also called criterion variables) must be made.

#### 2.3.3 Random Forest

Random forest is another flexible supervised machine learning algorithm used for both classification and regression purposes. The "forest" references a collection of uncorrelated decision trees, which are then merged together to reduce variance and create more accurate data predictions.

# 3 Methodology and Results

In this project, the main objective was to find who are nurses that accept whatever shift we show them? Who are the nurses who only work when they find the exact shift they like? Which nurses can work more shifts in a week and which nurses cannot? Who are the weekend warriors? Who are the holiday heroes? And to improve in-app shift recommendations, and offer more effective promotions if understand what behavior patterns characterize a given nursing professional by accomplish the previous questions.

## 3.1 Data Processing and Data Visualization

In this project, we have three data sets which are accept behavior, nurse info and app behavior. The accept behavior data gives timestamps for every time a nurse has accepted a shift, as well as the start time of that shift. If the nurse has "released" or canceled the shift, there's a timestamp for that event as well. The "client-id" designates the facility where the shift is taking place. There are some well-known behaviors in this data set alone. Most nurses are consistent with their preferences for weekdays vs weekends and mornings vs afternoon vs nights. Some nurses spread their time across several facilities while others consistently return to the same facility for several months. Some nurses release a lot of shifts - they can do so without any penalty from us if they release $> 72$ hours in advance. Releases $< 72$ hours incur various non-financial penalties and eventually lead to termination if there are too many in a short window. The nurse info data gives some basic info about each nurse, designated by their "pid" (user id). All nurses in our sample have completed at least 5 shifts. There are timestamps for their application date, as well as the day they first accepted a shift, the date they first completed (not accepted) a shift, etc. And app behavior is gives summary info on each app browsing session - specifically how many shifts were viewed and clicked. You might see several browsing sessions in the same hour for the same pid. Other nurses might go days or weeks between sessions. The data only spans from mid Aug 2020 to the present, so we'll have to assume that whatever patterns we detect in that window apply elsewhere.

We focused on the accept behavior at first. As we can see from the data set, it has the release date for the shifts. This means that the user canceled their shifts before the the shifts start. And NA in our data set means the user did not cancel the shift. Therefore, we need to drop all released shifts by pid (user). After that, we generate the plots in the figure 1:

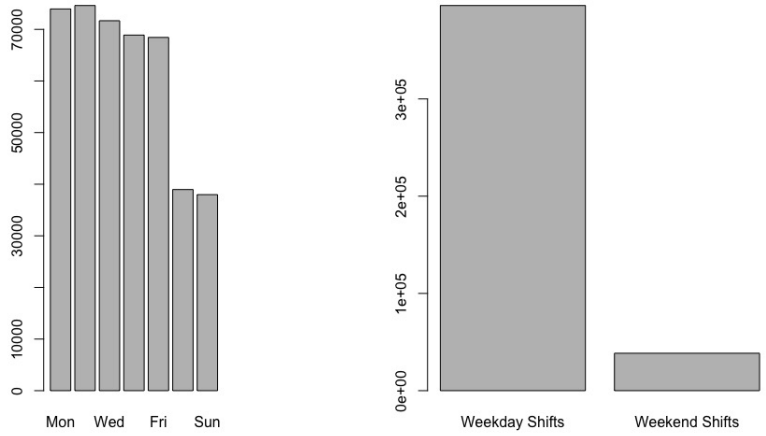


Figure 1: All shifts on Weekdays and Weekends without released data. At the right part is comparison.

From figure 1 we can see that most users tend to accept shifts in weekdays. Especially on Monday and Tuesday, we can see these two days has most shifts. On the right hand of the figure is the comparison of the weekday shifts and weekend shifts. We can conclude that most users prefer to accept shifts on weekdays.

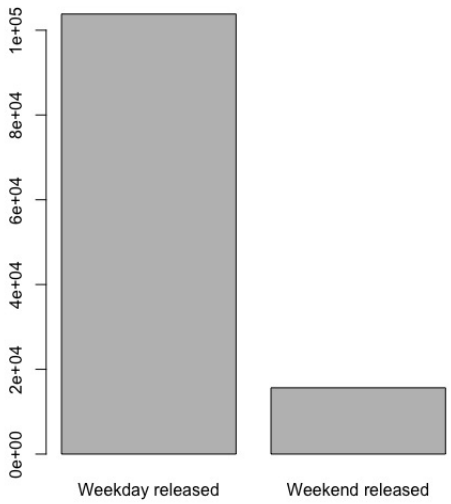


Figure 2: Comparison of Weekday release and weekend released

Based on the figure 2, we conclude that there are approximately 86% people canceled their shifts on the weekdays.

Then we plot total shifts have been accepted in the the previous years by every month which is shown in Figure 3.

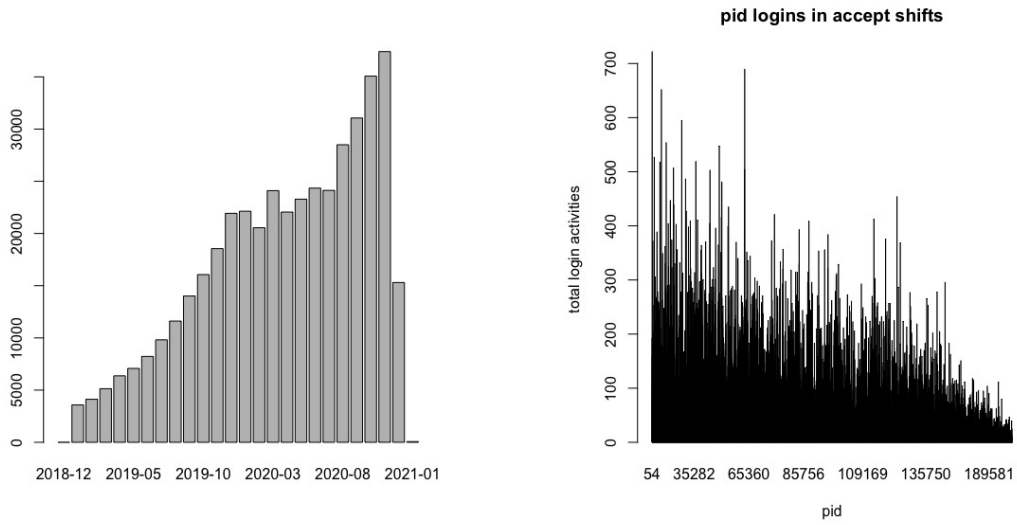


Figure 3: Weekly shifts and login activities

In Figure 3 we can see there is a significant increasing trend from December 2018 to December 2020. At the right hand, it is the total login activities for each pid by order.From these app behavior graphs, most of the shifts fall under the weekday. Interestingly enough, there are 2.5 times more weekdays than weekends however there are about about 2x shifts greater on the weekday than the weekend. Therefore there is an uneven distribution of weekend vs weekday shifts. The weekend shifts are the most busy days on the app, and ones where there are the most amount of shifts whereas Wednesday there are the least amount of shifts.

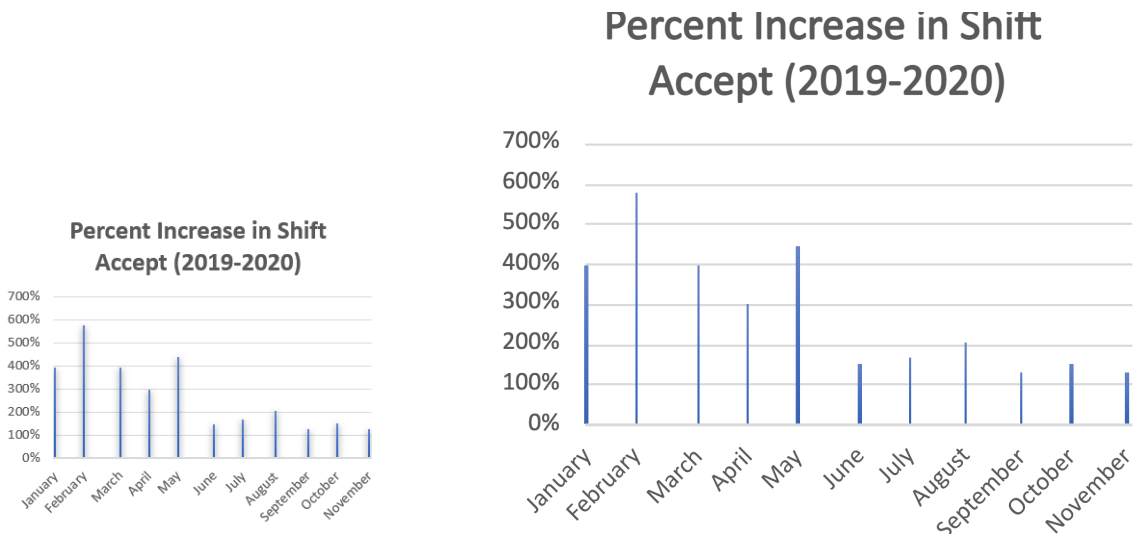


Figure 4: Percent Increase in Shift Accept

Figure 4 shows the increase in shifts accepted from 2019 to 2020. Note December is omitted because there was no data for December of 2020. As seen from the graph, every month has an increase in accepted shifts. This may be due to two reasons. One, there are more nurses that are using the app. Two there are more desirable shifts available that has a greater positive appeal to nurses using the app. Figure 4 shows the increase of shifts from the previous year from 2019 to 2020. Every month had a positive increase in shifts available on the app. The most significant increase in available shifts are in January. Two plots in figure4 look very similar because both graphs use the accept behavior data set. In this data set, it is very common that nurses are accepting shifts that are in the same month, unless it is nearing the end of the month which then the date of the shifts will be for the upcoming month. This shows that the shifts on the app are likely

not more than a month in advance.

Figure 5 shows the logins per pid over the span of the entire data set. The most important information is that this graph offers is that there is a wide range of active and inactive users. A mean of 178 can be used as a basline to differentiate between active and inactive users. This data would need to be further filtered out for more critical pieces of information to be found. Perhaps future steps would be to average out the logins per pid based on a monthly or weekly basis and analyze that information. Or accumulate a average logins per month to see if users are consistently using the app over a long period of time or if the users are excessively using the app in a month and inactive for the rest of the year. We think this would critically filter out this graph into a more digestible medium.

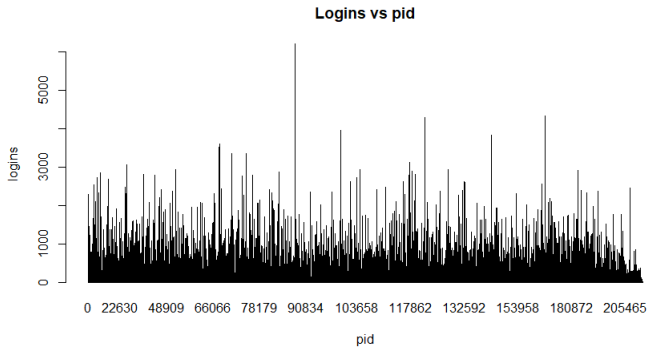


Figure 5: Mean: 178.0315 Max: 6204 Min: 1 Std: 331.0902

## 3.2 Clustering Methods and Dimensional Reduction

### 3.2.1 Principal Component Analysis

PCA was run on the nurse info data set we combined three data sets with all observations which has a unique PID. These observations included the varibles:

- Years of Prior Work History.
- Prior Distinct Jobs.
- Total Accepted Shifts.
- Total Viewed Shifts.
- Total Clicked Shifts.
- Day difference between date of first shift and date of starting app. (date.diff)
- Day difference between fifth and first shift.
- Average day difference between shift date and accepted date.
- Difference between first and last app open.
- Clicks per view.

We can can see the PCA analysis biplot as following in figure 6. This plot show the results of the PCA. The 1st PCA is on the x-axis and the 2nd PCA is on the y-axis. There are two correlations identified from this method of analysis. The first one being the correlations between the variables as shown from the PCA1 correlations of Total Viewed Shifts, Total Clicked Shifts, Total Accepted Shifts, Days between first and last app opening. The total shifts accepted is more correlated with the duration of time they spend using the application seen by the correlations with how many shifts that the nurse clicks on and views. These variables can be used to classify the nurses' experiences using the app. Prior work history years, and Prior distinct jobs are the other variables that are correlated from this PCA. These can be useful in breaking down the experiences of nurses and their behaviors.

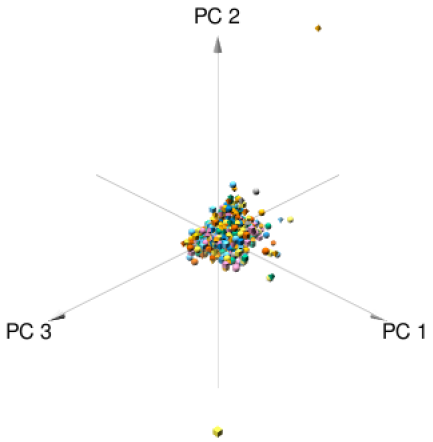


Figure 6: PCA Analysis

### 3.2.2 K-mean clustering

After we use PCA method for our combined data set, we apply two clustering techniques to the data. The first one we used is K-mean clustering method. The result is shown in figure 7.

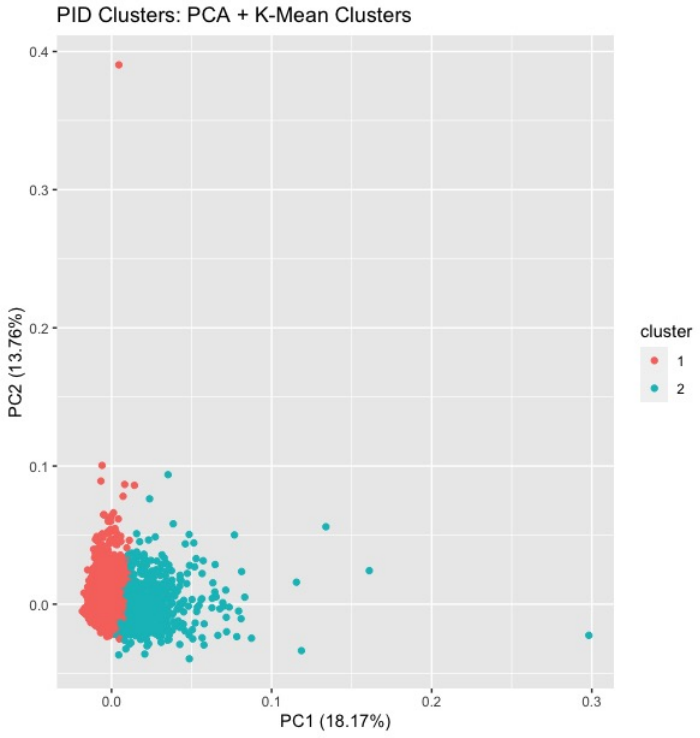


Figure 7: PID Clusters: PCA + K-mean Clusters

As we can see in the figure 7, we choose to separate the data into two clusters. In the whole combined data set we created, there are 6846 observations in total. After applied K-mean clustering algorithm, we labeled 5606 nurses. And in the second cluster, we have 1240 nurses.

### 3.2.3 DBSCAN clustering

To apply DBSCAN, we need to decide on the neighborhood radius eps and the density threshold minPts. The rule of thumb for minPts is to use at least the number of dimensions of the data set plus one. To achieve this step, we need to apply a k-NN distance plot to find out the eps. As we can see on the left side of the figure 8, we can find out the eps is around 2-3. In this case, we choose $eps = 2$ and $minPts = 3$. After that, we applied the $fpc :: dbscan()$ function in R and get the plot on the right in figure 8. The black points are outliers and one of the pros of using DBSCAN algorithm is we do not need to choose the cluster numbers by hand. In the other words, the computer will auto-generate the clusters. And in this case, we got 5 clusters.

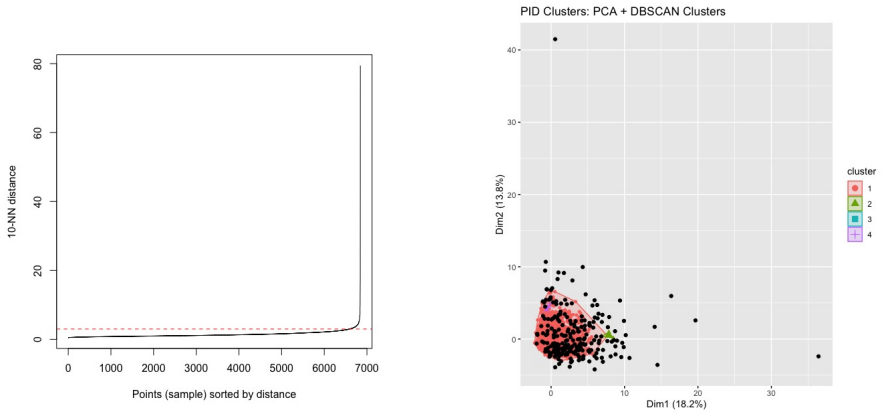


Figure 8: PID Clusters: PCA + DBSCAN Clusters

## 3.3 Determining App-Usage

A major goal is to identify when nurses begin using the app less. If this can be determined in a timely manner, the behavior of the application can be adjusted to draw an individual back to the application.

To do this, for each nurse, the difference between each of their accepted shifts was found and the value was averaged. After finding the average time between accepted shifts, the time since their last login can be used in comparison to determine which nurses have been off the application for an extended period of time.

It is of interest to find the days of the week that each nurse's behavior falls on most. This may be applied to app logins, shift views, and shifts accepted. We created a histogram for this and it is shown in figure 9.

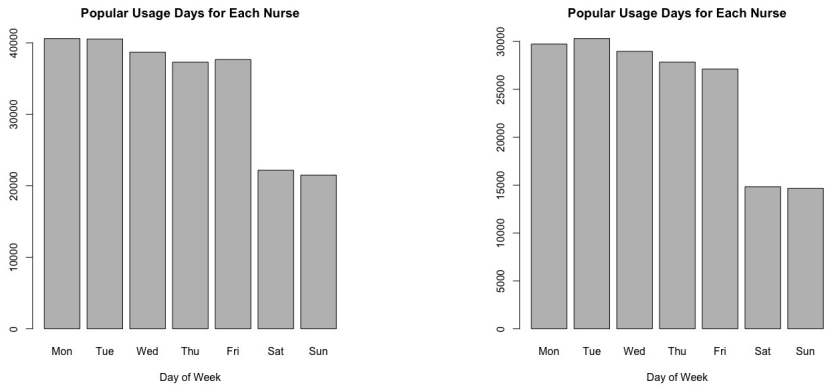


Figure 9: Popular Usage Days for Each Clusters

Based on the figure 9, we can clearly see that, most users tend to use the app during the weekdays.

## 3.4 How Long are Nurses Planning Ahead?

We think this question is quiet important in this study. This is because that IntelyCare wants to knwo that some nurses behavior, such as who are nurses that accept whatever shift we show them? Who are the nurses who only work when they find the exact shift they like? Which nurses can work more shifts in a week and which nurses cannot? Who are the weekend warriors? Who are the holiday heroes? These behavioral cohorts will allow them to personalize the nurse experience in important ways. We can send more timely and relevant communications, improve our in-app shift recommendations, and offer more effective promotions if we understand what behavior patterns characterize a given nursing professional. Therefore, by classifying the nurses based on the cluster data to find out who would like to planning the shifts ahead is very important. This can help the nurse to schedule the shifts better and can help IntelyCare to provide a personalized schedule to these nurse before some important days such as holiday, weekends etc.

Based on the left hand plot in figure 10, we can see that the user plan ahead days' range is from 0 to 25 days in the first cluster. And we calculated the average days difference that they schedule the shifts ahead of the shift which is 4.57 days. For the right hand side plot is for second cluster in figure 10. We can see that the user in this cluster has a longer range which is from 0 to 40 days. And based on the plot, we can conclude that most users are willing to arrange their working hours 0 to 10 days in advance. And average days they would like to arrange their working hours is 6.71 days.

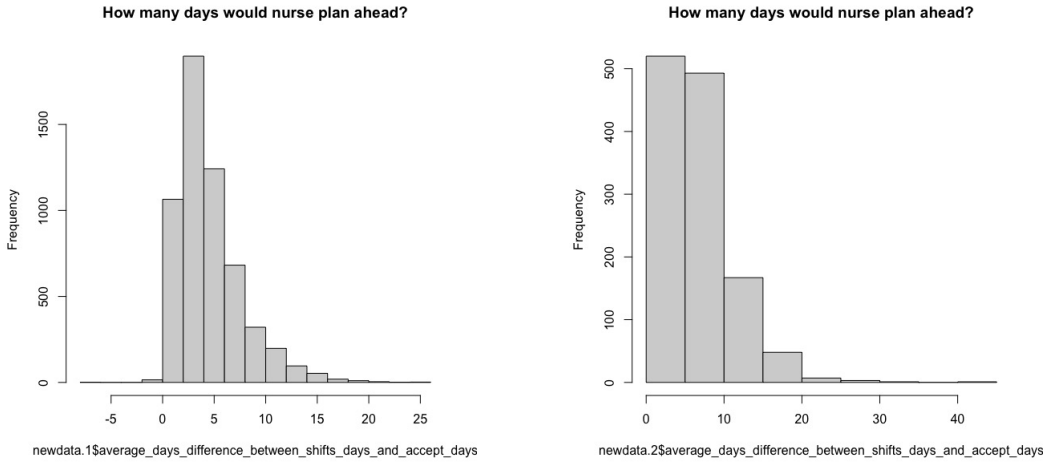


Figure 10: Cluster1 vs Cluster2

Then, we need to classified the users into different groups by using the clusterd data set. We created binary numbers to represent each groups. In the following part, group 0 will represent users that not tend to planning ahead for their schedule for the median of the day difference data which is 4.10 days. In this study, we developed several models to predict and classify the users and their behaviors.

### 3.4.1 Logistic Regression Results

As we mentioned in the previous chapter, logistical regression is selected when the dependent variable is categorical, meaning they have binary outputs, such as "true" and "false" or "yes" and "no.". Therefore, we come up our first question what is the probability that a nurse willing to schedule their shifts greater than the median number which is 4.10 days.

First of all, we chose the clustered data set by *K-mean clustering* results. In other words, we have two different clusters for our data set. Secondly, we subset the data based on the clustered data we get and add the cluster into data set for each pid. For the first cluster, we have 5600 users and the second cluster we have 1200 users. Then, we randomly generate 3000 observations for the testing group in the first cluster, and 300 observations for the test group in the second cluster.

We trained a training data set for each cluster and test the model on the testing group we randomly generate. We created a confusion matrix, we can clearly see in the figure 11, our model is able to classify most observations into correct groups and we can see the accuracy of the model is high as well which is 99.7% for the data in the first cluster and 99.67% for the data in the second cluster. Moreover, to check the

```
> table(glm_pred1,test_response1)
         test_response1
glm_pred1    0    1
        0 1580    2
        1    6 1412
> mean(glm_pred1==test_response1)
[1] 0.9973333
```

```
> table(glm_pred2,test_response2)
         test_response2
glm_pred2   0   1
        0  87   0
        1   1 212
> mean(glm_pred2==test_response2)
[1] 0.9966667
```

Figure 11: Logistic Model for two clusters

model is a good model or not in our study, we used *pchisq* function in R as well. And we get the p-value is $2.754571_{*10^{-73}}$ and $3.550814_{*10^{-22}}$ respectively. We can conclude that since the p-value is smaller than the landmark which is 0.05, we can say the two logistic models we built is significant in this study.

### 3.4.2 Linear Discriminant Analysis Results

Linear Discriminant Analysis (LDA) is most commonly used as dimensionality reduction technique in the pre-processing step for pattern-classification and machine learning applications. We are using the same data set as we did for logistic regression. And first of all, we can check the following figures which are some results we built for the two LDA models in two clusters.

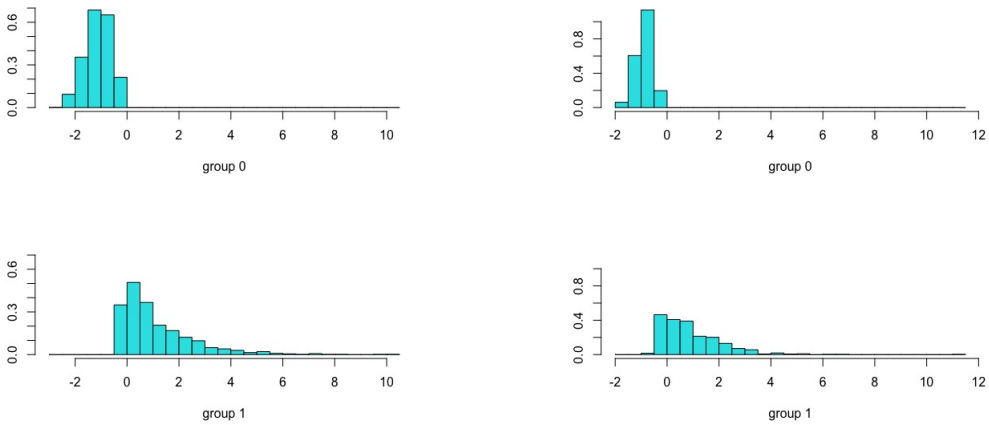


Figure 12: LDA histogram results for each group in cluster 1 and cluster 2

```
> table(lda.class1,test_response1)
          test_response1
lda.class1    0    1
         0 1586  313
         1    0 1101
> mean(lda.class1==test_response1)
[1] 0.8956667
> mean(lda.class1!=test_response1)
[1] 0.1043333
```

```
> table(lda.class2, test_response2)
          test_response2
lda.class2   0   1
         0  81   1
         1   7 211
> mean(lda.class2==test_response2)
[1] 0.9733333
> mean(lda.class2!=test_response2)
[1] 0.02666667
```

Figure 13: Confusion Matrix for LDA in first cluster and second cluster

Based on the figure 12 and figure 13, we can see the LDA model can classified the observations into each correct groups as well. However, by looking at two confusion matrix we created, we can see that it has a lower accuracy compare to the logistic model we created, but it still has 89.5% for the first cluster and 97.33% for the second cluster.

### 3.4.3 Random Forest Results

Random forest is another flexible supervised machine learning algorithm used for both classification and regression purposes. The "forest" references a collection of uncorrelated decision trees, which are then

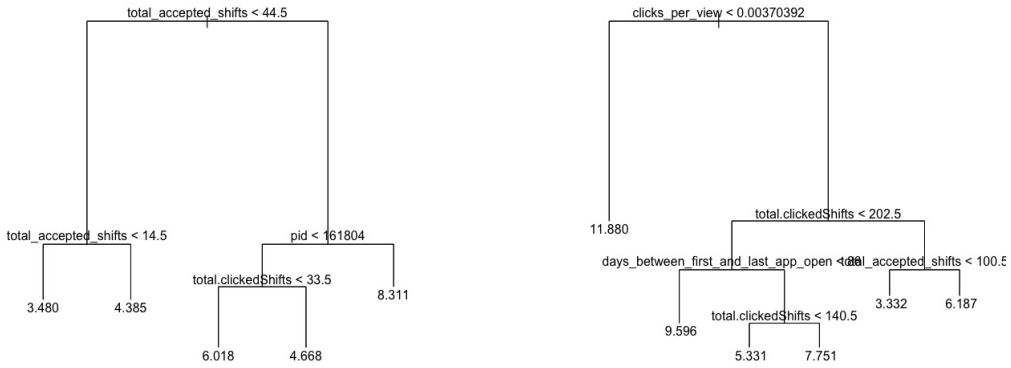


Figure 14: Prune Trees Results for cluster 1 and cluster 2

merged together to reduce variance and create more accurate data predictions.

First of all, in this method, we built two models based on the clustered data which is regression tree. However, we come up a different question. We want predict the how long will the nurse arrange their shifts ahead of the shift's date. In our data set, we have the data for the average days difference for each nurse, therefore, in this case, we chose this variable as our response variable. We got two trees for the two clustered data. Then, we did cross-validation for each trees and get the following figures.

To get the accuracy of the regression tree model, we need to find the mean square error (MSE) for each model. The MSE is a measure of the quality of an estimator—it is always non-negative, and values closer to zero are better. MSE can calculate by the following formula:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2$$

In our study, we get prune trees' MSE are 9.693265 for the first cluster and 22.46222 for the second cluster. Therefore, these two prone trees are really good model to predict the nurse behaviors.

After that, we built the random forest model based on this. The MSE we get for both models are 4.526751 and 8.87655 respectfully.

# 4 Conclusions and Applications

Our most conclusive results include; nurse behavior on weekends compared to weekdays, app usage frequency, holiday behavior, dormant periods, and overall increase in app usage.

A massive goal in tailoring app experience is to cut down on the amount of things the user needs to click on. One way this was solved, was determining what days nurses are accepting shifts, and what days do these shifts occur on. Effectively, this means that IntelyCare can create custom functionality such as, the first time a user logs on after the week begins, their previous shift, or the closest shift to it, can be promoted to them. The more data on the nurses typical behavior regarding when the shifts usually occur, the more accurately their desires can be predicted, and the use will spend less time in the app and more time accepting shifts. This ultimately leads the customer to associating a better feeling with the app, knowing it is an extremely easy experience, leading to them using it more.

The results, are that nurses are accepting shifts early on in the week, but these shifts are occurring on the weekends. This was determined by simple x-y plots of various columns inside the accept behavior data. Also, it was determined that when a nurse typically accepts shifts in the afternoon- that is a harder habit, or constraint, to break than which day they are accepting the shift. This was mentioned during conversation with IntelyCare although it is extremely important to note, as analysis could be done further with the provided data.

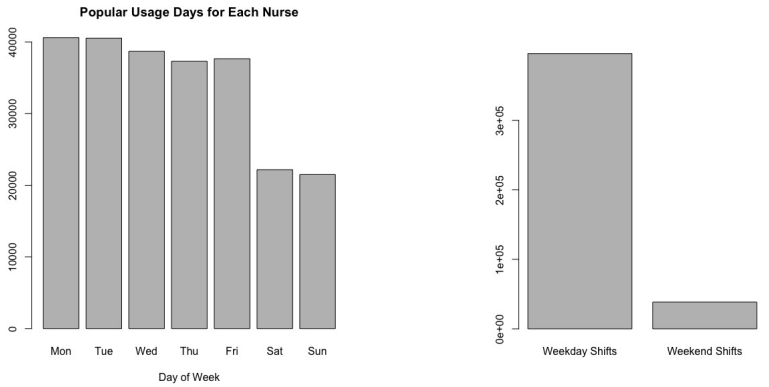


Figure 15: Day that shifts fall on (left) versus day that nurses accept the shift (right)

It was determined that the average day difference, between each accepted shifts, helps determine if a user has stopped using the app for longer than usual. For example, if a user typically uses the app once every two weeks, it should be no surprise if the user has not used the app in a few days. But, when this analysis is dynamically applied, a user that has not used the app in a week, who usually uses it three times a week, can be flagged. The user may then me prompted back to the application with something like a notification. At the time of calculation, data was skewed because the time since last login was offset by the amount of days since the data was last recorded. An arbitrary date was then chosen to find the difference from, to prove both the concept and the code written to accomplish this. This proved to be useful, as the results were expected- returning the ID's of users who have been dormant for longer than usual, and if desired, returning the average day difference, as well as the current day difference.

It was learned early, that behavior around a holiday was different from typical days. This means that there is probably a predictable difference in data during these special occasions. The most predictable behavioral difference was that nurses are planning ahead for these shifts much more. With shifts on non-holidays averaging six days of planning, shifts that fall on holidays are being planned ahead on average of eight days ahead. This is meaningful because clearly there is more demand for these shifts since nurses are picking them up more ahead of time. When determining application features that apply to this, it is likely that when promoting a shift with a holiday characteristic, the shift should be promoted earlier so that it is a useful promotion to the user and they did not already accept the shift days beforehand. Therefore, we used some supervised learning techniques we mentioned in the previous chapters to predict and classified each nurse preference.

# 5 Reference


- Michael Hahsler, Matthew Piekenbrock, Derek Doran, "dbscan: Fast Density-based Clustering with R" `https://cran.r-project.org/web/packages/dbscan/vignettes/dbscan.pdf`